In the Claims

The status of claims in the case is as follows:

1. [Previously presented] A method for control and management of communication traffic, comprising the steps of:

expressing access rules as filters referencing system kernel data;

for outbound processing, determining source application indicia;

for inbound packet processing, executing a look-ahead function to determine target application indicia; said look-ahead function being executed within a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as non-deliverable, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

responsive to said source or target application indicia, executing filter processing; said filter processing including constructing and evaluating logical expressions of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

2. [Currently amended] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of executing said determining and executing steps within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet.

3. [Currently amended] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a task or thread identifier;

based on said task or thread identifier, determining a process or job identifier; and

based on said process or job identifier, determining job or process attributes for filter processing.

4. [Currently amended] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and said filter processing including the steps of:

determining a user identifier; and

based on said user identifier, determining user attributes for filter processing.

5. [Original] The method of claim 3, further comprising the step of determining from said task identifier a work control block containing said process or job identifier.

6. [Canceled]

7. [Canceled]

8. [Currently amended] The method of claim 1, wherein said protocol stack is a TCP/IP protocol stack, and further

comprising the steps of:

delivering to said filters infrastructure access rules for defining security context.

9. [Original] The method of claim 8, said infrastructure including logging, auditing, and filter rule load controls.

10. [Previously presented] A method for control and management of aspects of communication traffic within filtering, comprising the steps of:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel;

for an inbound IP packet, executing a look-ahead function within a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as non-deliverable, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and executing filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack; said filtering code constructing and evaluating logical expressions of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

11. [Original] The method of claim 10, said non-IP packet data including context data regarding said IP packet.

12. [Original] The method of claim 10, said non-IP packet data including data specific to a task generating said non-IP packet data.

13. [Original] The method of claim 10, said non-IP packet data including data specific to a task that will receive said IP packet.

14. [Original] The method of claim 11, said context data including packet arrival interface indicia.

15. [Canceled]

16. [Canceled]

17. [Canceled]

18. [Previously presented] A method for centralizing system-wide communication management and control within filter rules, comprising the steps of:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values;

for an inbound packet, executing a look-ahead function within a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as non-deliverable, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered by said sockets layer;

said selector referencing data that does not exist in IP packets;

processing said filter statements, including constructing and evaluating logical expressions of arbitrary length, and selectively using a set of logical operators, alternative filter selector fields, and value set.

19. [Currently amended] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said parameters selectively including userid, user profile, user class, user group, user group authority, user special authority, job name, process name, job group, job class, job priority, other job or process attributes, and date & time.

20. [Currently amended] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and said filters statements being provided within a user interface to said system.

21. [Currently amended] The method of claim 18, wherein said protocol stack is a TCP/IP protocol stack, and further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel;

said filtering code accessing filtering attributes further limiting traffic selectively to job indicia; and

operating said filtering code within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said traffic.

22. [Currently amended] A method for traversing a portion only of a protocol stack to disallow selective IP packet traffic, comprising the steps of:

receiving a packet in the kernel of the operating system of a first node from an application, said kernel including a filter processor; said filter processor for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set;

for inbound packet processing to a first node from a second node, executing a look-ahead function in the system kernel of said first node to determine a target application; said system kernel including a TCP/IP protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as non-deliverable, and receives back from said transport layer indicia identifying the application layer application to which said packet would have been delivered;

for both said inbound packet processing, and for outbound packet processing from said first node to said second node, executing within said kernel the steps of

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling communication traffic.

23. [Previously presented] A method for expressing access rules as filters, comprising the steps of:

providing a filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets for controlling access to an application;

for an inbound IP packet, executing a look-ahead function within a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said IP inbound packet, said IP layer provides to said transport layer said inbound IP packet, marked as non-deliverable, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered; and

processing said filter statements by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set referencing said application layer application.

24. [Previously presented] A method for managing and controlling communication traffic by centralizing access rules in filters executing within and referencing data available in system kernels, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes; and

executing said filters by constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set.

25. [Previously presented] The method of claim 24, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node by executing a look-ahead function within a protocol stack including an IP layer, a transport layer, a sockets layer, and an application layer and which, for said inbound packet, said IP layer provides to said transport layer said inbound packet, marked as non-deliverable, and receives back from said transport layer indicia, provided to said transport layer by said sockets layer, identifying the application layer application to which said packet would have been delivered;.

26. [Canceled]

27. [Canceled]

28. [Canceled]

29. [Currently amended A method for managing and controlling communication traffic by centralizing the access rules, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node, said kernel including a filter processor for constructing and evaluating logical expressions of arbitrary length, said logical expressions selectively including a set of logical operators, alternative filter selector fields, and value set;

processing said packet within a TCP/IP stack;

by determining a task ID;

responsive to said task ID, determining a corresponding work control block;